

# ESGF + DOCKER

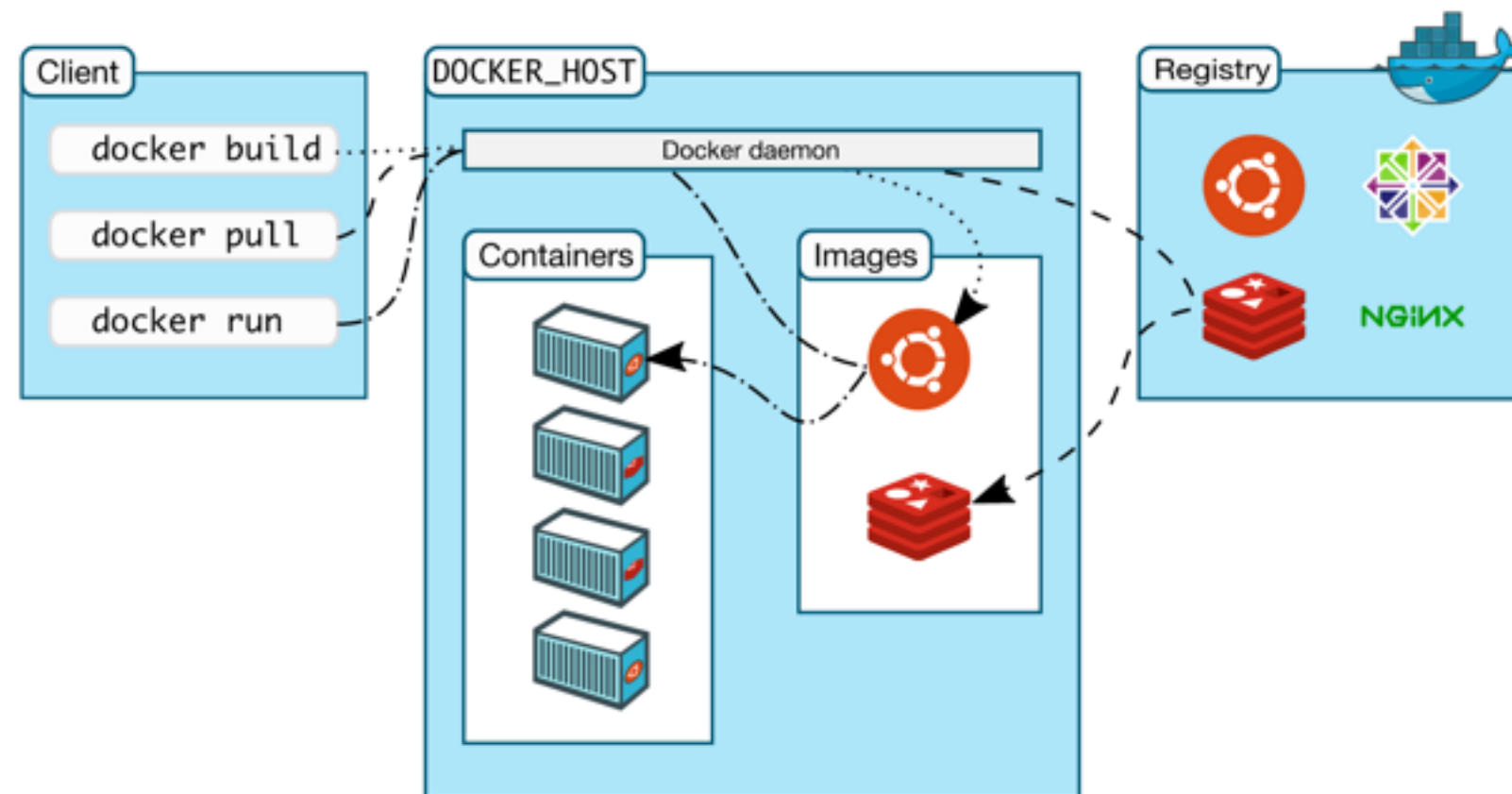
ESGF F2F Workshop,  
Washington, DC, December 2016

**Luca Cinquini**

NASA/Jet Propulsion Laboratory + California Institute of Technology

# Docker in a Nutshell

- Docker is the leading “containerization” technology: run an application as a “black box” on any Docker-enabled server
  - ▶ Build: images are built as bundles that include the application itself, all of its dependencies, and “just-enough-operating-system”
  - ▶ Ship: images are hosted on online repositories such as DockerHub
  - ▶ Run: images are run as containers on any host that includes a Docker daemon

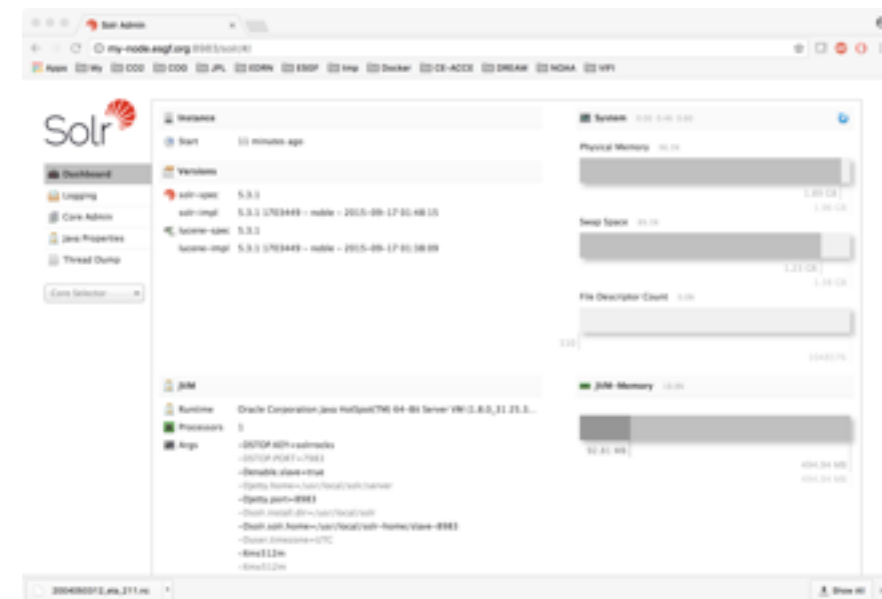
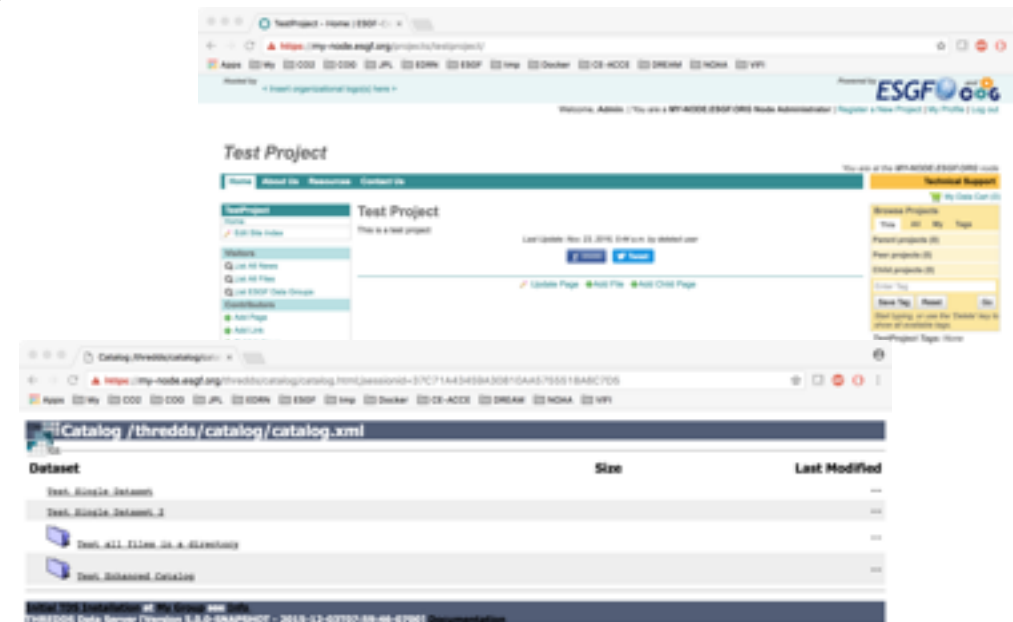
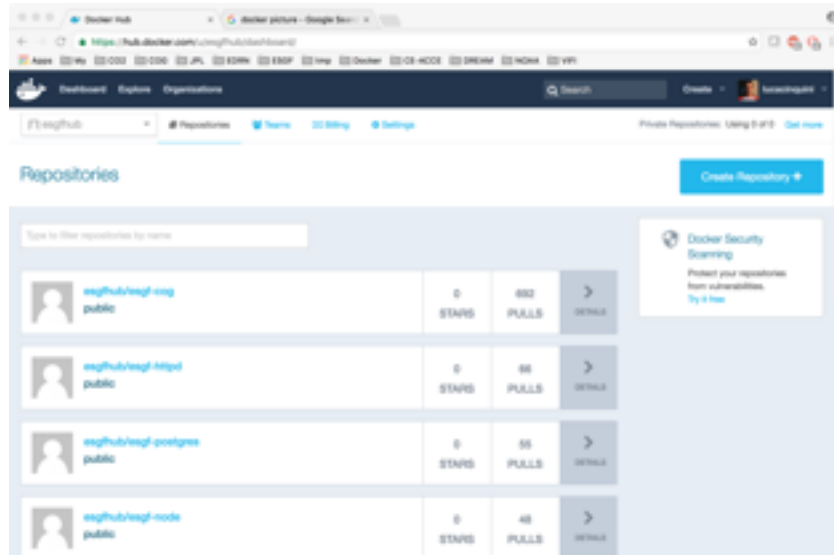


- Why using Docker for ESGF ?
  - ▶ Promises to greatly improve installation and maintenance of an ESGF node
  - ▶ Part of DREAM strategy for modularizing the ESGF architecture and porting it to other domains

- Pre-requisite: install Docker Engine on host (Linux, MacOSX, Windows)
- Instructions:
  - ▶ git clone <https://github.com/ESGF/esgf-docker.git>
  - ▶ cd esgf-docker
  - ▶ export ESGF\_HOSTNAME=<host FQDN>
  - ▶ export ESGF\_CONFIG=<some directory>
  - ▶ ./esgf\_node\_init.sh
  - ▶ docker-compose up

<https://my-node.esgf.org/>

<https://hub.docker.com/esgfhub/>

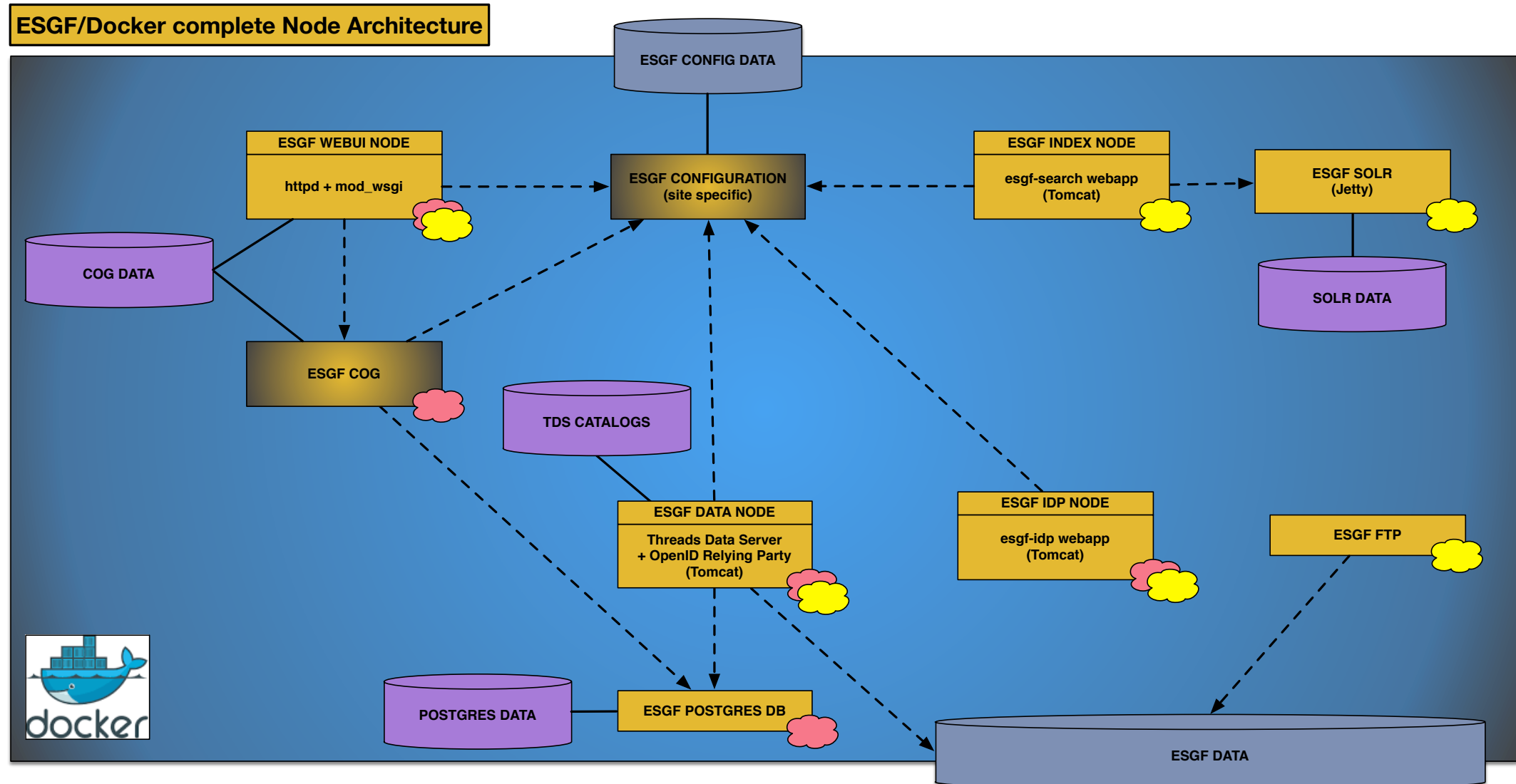


CoG

TDS

Solr

`docker-compose -f docker-compose.yml up`



- All ESGF services are run as independent, interacting Docker containers
- Specific Node configuration (certificates, passwords, XML files) saved in `$ESGF_CONFIG`
- Specific Node data (Postgres db, Solr indexes, TDS catalogs, NetCDF files, CoG site media) stored on Docker volumes
- Custom networks isolate applications for additional security (for example, Postgres db)

Dockerfile:  
“recipe” for building a Docker image

docker-compose.yml: configuration file  
for bundling several images

```
node — vi Dockerfile — 80x44
# Docker image based on Centos 6.7 that includes:
# Oracle Java 8
# Python 2.7 + a few common Python packages
# Also includes common, non-site specific ESGF env variables (ESGF_HOME).

FROM centos:centos6.7

MAINTAINER Luca Cinquini <luca.cinquini@jpl.nasa.gov>

# install dependencies with yum
RUN yum -y update && \
    yum groupinstall -y development && \
    yum install -y \
    yum-utils \
    bzip2-devel \
    git \
    lsof \
    which \
    hostname \
    openssl \
    openssl-devel \
    sqlite-devel \
    sudo \
    tar \
    wget \
    zlib-dev \
    sqlite-devel \
    freetype-devel \
    postgresql-devel \
    libjpeg-turbo-devel && \
    yum clean all

# install Oracle Java 8
ENV JAVA_VERSION 8u31
ENV BUILD_VERSION b13
RUN wget --no-cookies --no-check-certificate --header "Cookie: oraclelicense=acc
ept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/$JAVA_VERS
ION-$BUILD_VERSION/jdk-$JAVA_VERSION-linux-x64.rpm" -O /tmp/jdk-8-linux-x64.rpm
RUN yum -y install /tmp/jdk-8-linux-x64.rpm && \
    rm /tmp/jdk-8-linux-x64.rpm
RUN alternatives --install /usr/bin/java jar /usr/java/latest/bin/java 200000
RUN alternatives --install /usr/bin/javaws javaws /usr/java/latest/bin/javaws 20
0000
```

```
esgf-docker — vi docker-compose.yml — 80x44
# required env variables:
#
# ESGF_CONFIG : points to the root ESGF configuration directory
#   example: export ESGF_CONFIG=/Users/cinquini/ESGF_CONFIG
# ESGF_HOSTNAME: HostName or IP address for apache httpd front-end server
#   example: export ESGF_HOSTNAME=my.esgf.node
#   example: export ESGF_HOSTNAME=`docker-machine ip`
# ESGF_DATA_DIR: root of ESGF data directories

version: '2'

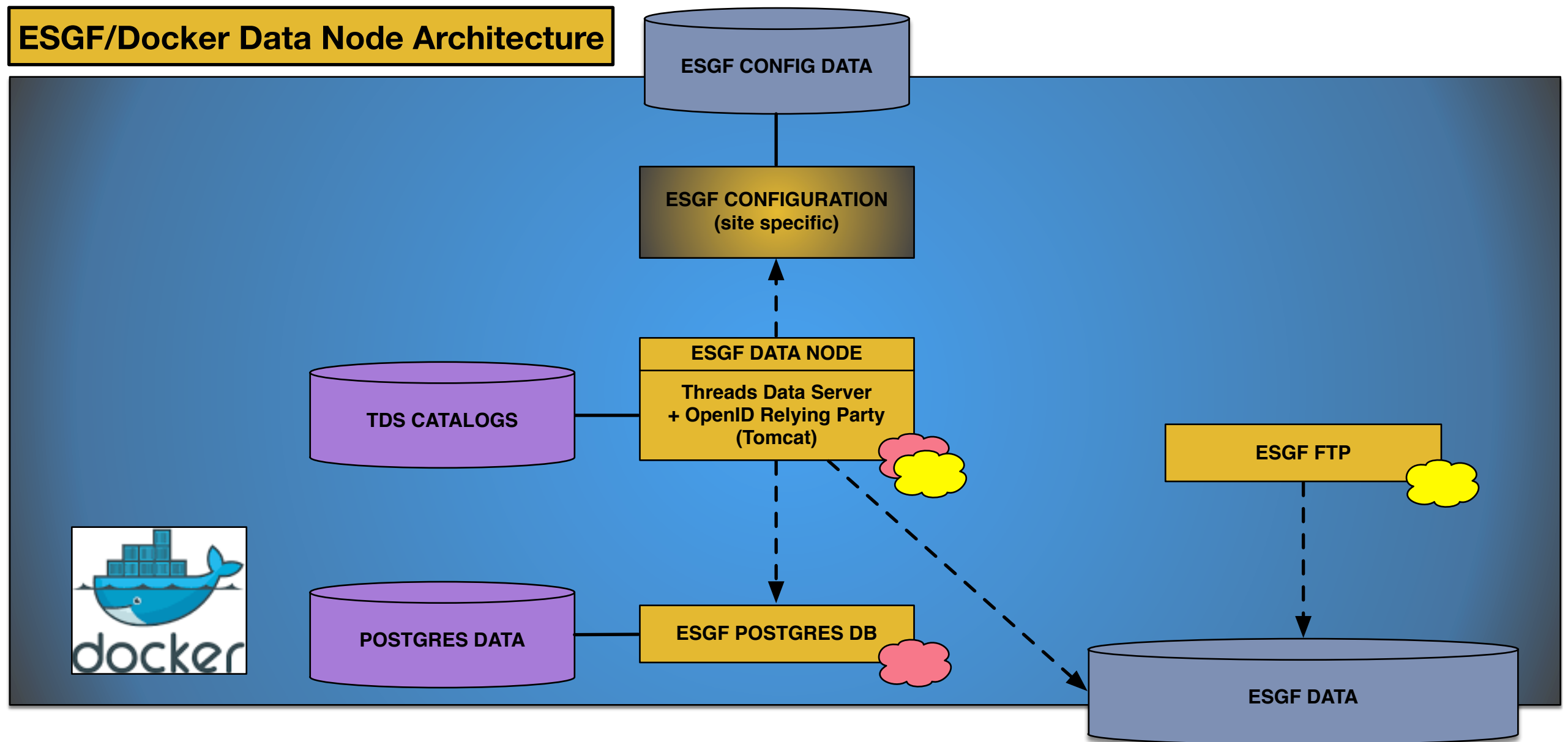
networks:
  # network to keep postgres database isolated
  dbnetwork:

services:
  # ESGF Data Node
  esgf-data-node:
    image: esgfhub/esgf-data-node
    container_name: data-node
    expose:
      - "8080"
      - "8443"
    ports:
      - "8080:8080"
      - "8443:8443"
    networks:
      - default
      - dbnetwork
    volumes:
      - tds_data:/esg/content/thredds
    volumes_from:
      - esgf-config
    environment:
      - ESGF_HOSTNAME
    depends_on:
      - esgf-postgres

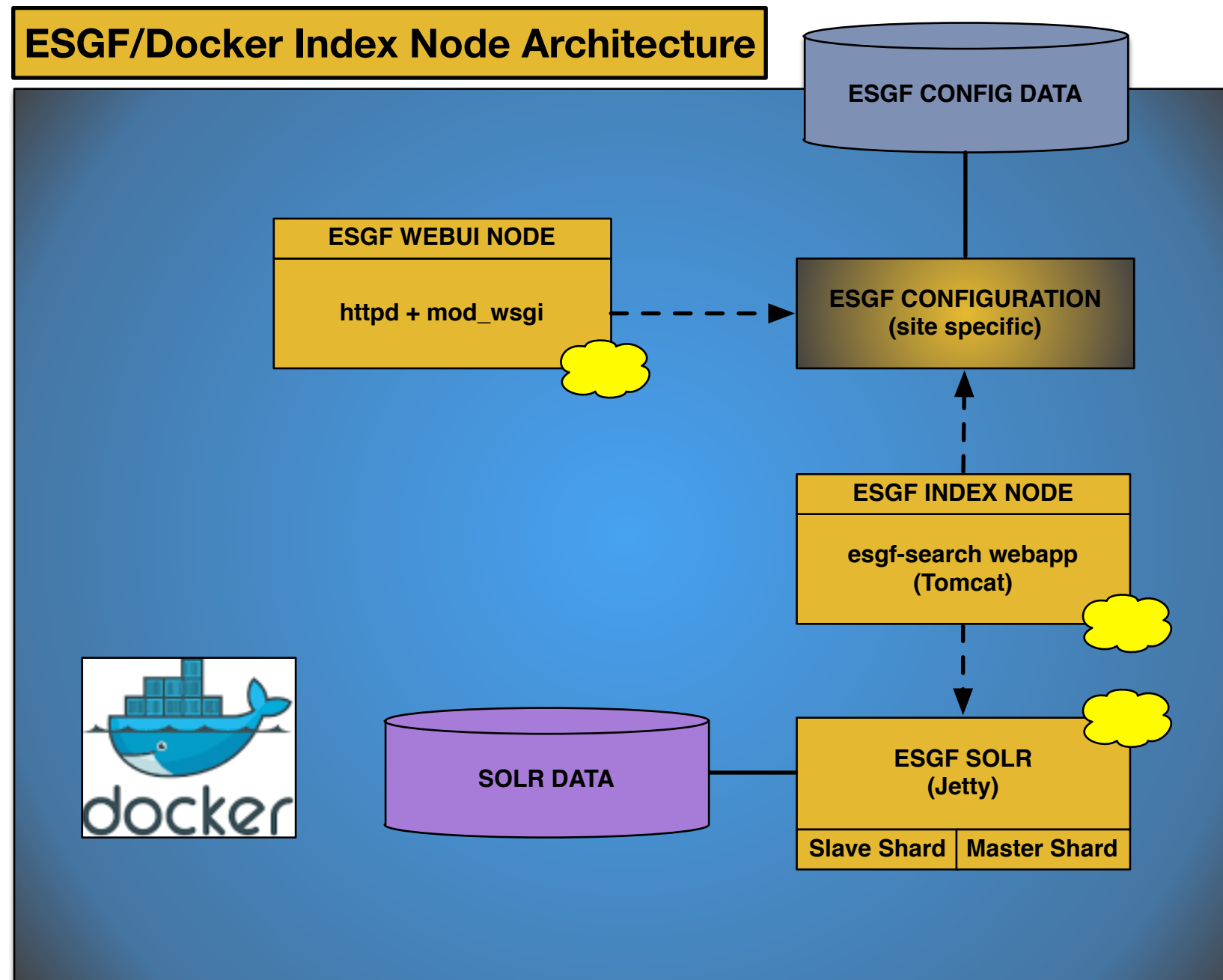
  # ESGF Identity Provider
  esgf-idp-node:
    image: esgfhub/esgf-idp-node
    container_name: idp-node
    "docker-compose.yml" 190L, 4544C
```



`docker-compose -f docker-compose-data-node.yml up`

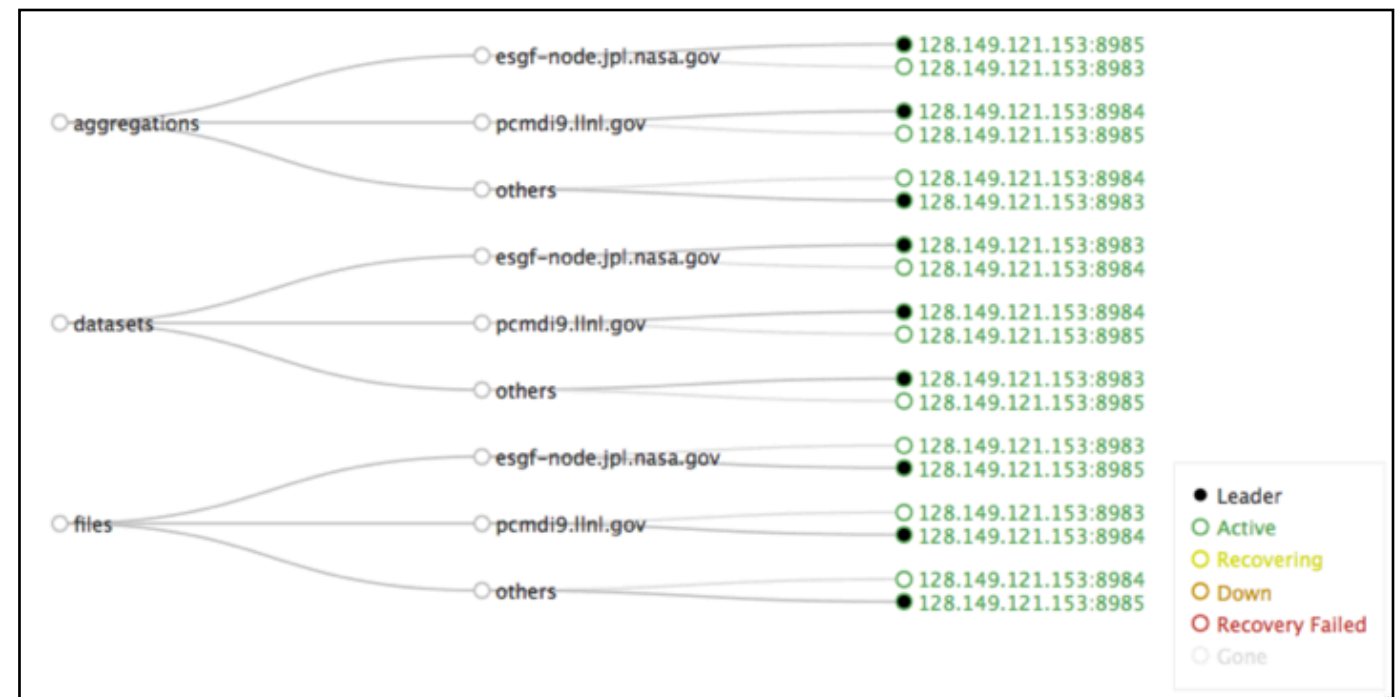
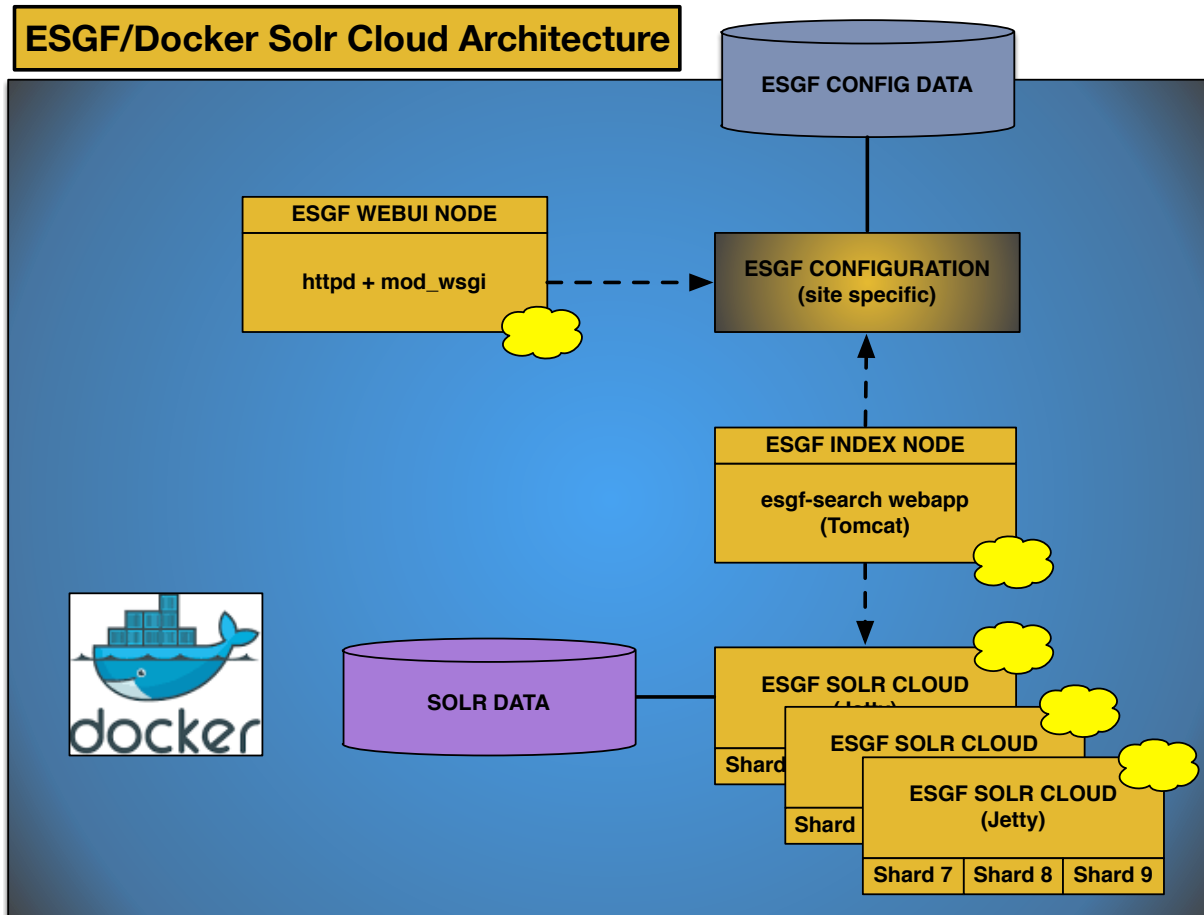


```
docker-compose -f docker-compose-index-node.yml up
```



## Using standard Solr replication + distributed search

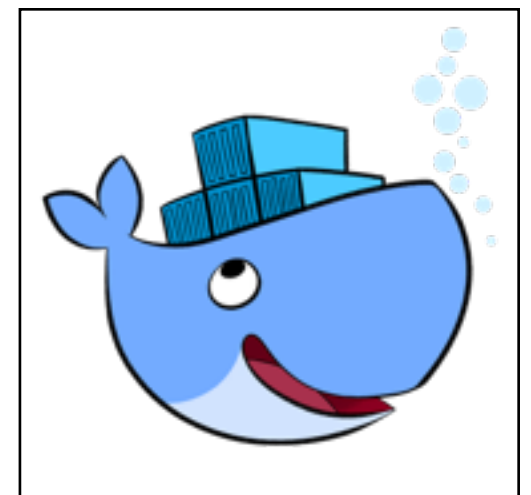
`docker-compose -f docker-compose-solr-cloud.yml up`



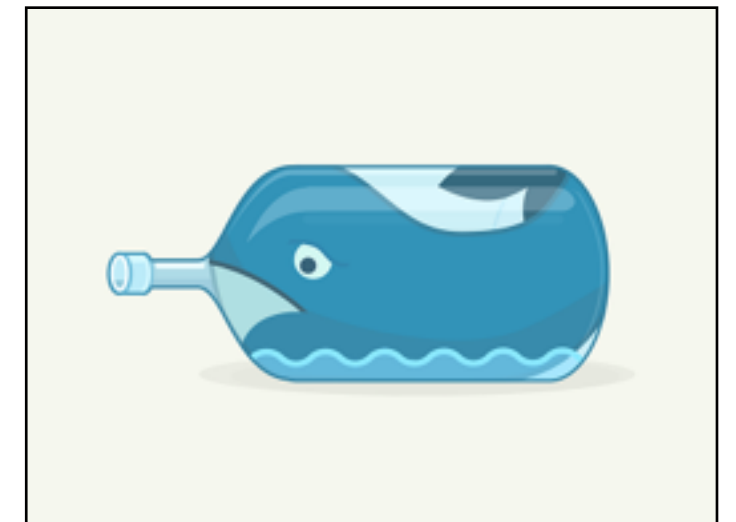
- Solr-Cloud advantages:
  - ▶ Automatic distributed indexing and searching (no custom configuration)
  - ▶ Load balancing and high availability
  - ▶ Fault tolerance
  - ▶ Scalability



- Installation:
  - ▶ Installation scripts are much more modular and much smaller
  - ▶ Installation process is much easier: simply download the images, no compilation involved
  - ▶ Easier upgrades, possibly one module at a time, and reversible
  - ▶ Everybody runs exactly the same software
- Architecture:
  - ▶ Can define and deploy new architectures by simply writing new configuration files
  - ▶ Can introduce new modules by simply writing & wiring new images (solr-cloud, nginx, ...)
- Portability: ESGF node will run on any platform (Linux, MacOSx, Windows) including Cloud
- Scalability: modules can be scaled arbitrarily by running more containers (e.g. TDS, WPS...)
  - ▶ Caveat: application must be written to enable distributed access to data
- Other Docker/Swarm advanced features:
  - ▶ Scalability onto multiple hosts clusters
  - ▶ High availability, fault tolerance, rolling updates, ...



- Docker is a new paradigm for building and running applications:
  - ▶ New knowledge for application developers
  - ▶ New training for node administrators
  - ▶ Excellent and up-to-date documentation is a must
- Must port the remaining ESGF modules to Docker:
  - ▶ ESGF Publisher Client
  - ▶ Globus
  - ▶ MyProxy or new OAuth server
  - ▶ Desktop + Dashboard
  - ▶ LAS
- Must develop a process to migrate all application data:
  - ▶ ESGF Postgres database (users and data)
  - ▶ TDS catalogs
  - ▶ Solr indexes
  - ▶ CoG Postgres database and site data



- Establish a “whale team” of experts to work on:
  - ▶ Test the current infrastructure
  - ▶ Port the remaining modules
  - ▶ Work on other outstanding tickets
  - ▶ Develop a testing suite
  - ▶ Develop data migration tools
  - ▶ Revise and expand the documentation
  - ▶ Re-use current esgf-iwt biweekly meetings ?
  
- Migrating to Docker would take 6-12 months
  - ▶ Must support current or upgraded installed in the meantime
  - ▶ Might want to switch after CMIP6 - looking into the long-term longevity of ESGF...

